Hive执行方式

Hive的hql命令执行方式有三种：

1、CLI 方式直接执行

2、作为字符串通过shell调用hive –e执行（-S开启静默，去掉”OK”，”Time taken”）

3、作为独立文件，通过shell调用 hive –f或hive –i执行执行

1、CLI 方式直接执行 2、作为字符串通过shell调用hive –e执行（-S开启静默，去掉”OK”，”Time taken”） 3、作为独立文件，通过shell调用 hive –f或hive –i执行执行

方式1

键入“hive”，启动hive的cli交互模式。Set可以查看所有环境设置参数，并可以重设。其他命令如，

Use database 选择库

quit/exit 退出Hive的交互模式

set –v 显示Hive中的所有变量

set <key>=<value> 设置参数

执行本地shell ：!<cmd> 交互模式下可执行shell命令，例如（查看linux根目录下文件列表："!ls -l /;"）

操作云命令：dfs < command> 交互模式下直接操作hadoop命令如 dfs fs –ls

Hql语句 执行查询并输出到标准输出

add [FILE|JAR|ARCHIVE] <value> [<value>]\* 增加一个文件到资源列表

list FILE 列出所有已经添加的资源

方式二

Hql作为字符串在shell脚本中执行，如

hive -e "use ${database};select \* from tb"

查询结果可以直接导出到本地本件（默认分隔符为\t）:

hive -e "select \* from tb" > tb.txt

如果需要查看执行步骤，则在命令前面添加

set –x

1

另外，在shell脚本中，字符串有两种定义方式:

1) 直接定义字符串对象：sql=”字符串”

2) 通过命令定义：sql=$(cat <<endtag 字符串endtag)方式可以将字符串复制给sql，执行hql命令的shell脚本如下：

####### execute hive ######

sql=$(cat <<!EOF

USE pmp;

set mapred.queue.names=queue3;

drop table if exists people\_targeted\_delivery;

create table people\_targeted\_delivery

( special\_tag\_id int,

cnt bigint

);

INSERT OVERWRITE LOCAL DIRECTORY '$cur\_path/people\_targeted\_delivery'

ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'

select special\_tag\_id,count(1)

from t\_pmp\_special\_user\_tags

group by special\_tag\_id;

!EOF)

############ execute begin ###########

echo $sql

$HIVE\_HOME/bin/hive -e "$sql"

exitCode=$?

if [ $exitCode -ne 0 ];then

echo "[ERROR] hive execute failed!"

exit $exitCode

fi

！

方式三

将hql语句保存为独立文件，后缀名不限制，可以用.q或者.hql作为标识：

A，这个文件在cli模式下，用source命令执行，如：source ./mytest.hql

B，在shell中执行命令，如：hive -f mytest.sql

Hive指定预执行文件命令“hive –i”（或叫初始化文件）

命令：hive -i hive-script.sql

在hive启动cli之前，先执行指定文件（hive-script.sql）中的命令。

也就是说，允许用户在cli启动时预先执行一个指定文件，比如，有一些常用的环境参数设置，频繁执行的命令，可以添加在初始化文件中，比如，

某些参数设置

set mapred.queue.names=queue3;

SET mapred.reduce.tasks=14;

添加udf文件

add JAR ./playdata-hive-udf.jar;

设置Hive的日志级别

hive -hiveconf hive.root.logger=INFO;

# Hive使用脚本加载数据

2016年12月03日 16:38:57 [H\_Hao](https://me.csdn.net/haoyuexihuai" \t "https://blog.csdn.net/haoyuexihuai/article/details/_blank) 阅读数：1402

 版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/haoyuexihuai/article/details/53445663

#### **方式一：直接写在脚本中**

load\_track\_logs.sh：

#!/bin/sh*## 环境变量生效*

. /etc/profile

*## HIVE HOME*

HIVE\_HOME=/opt/cdh-5.3.6/hive-0.13.1-cdh5.3.6

*## 日志目录*

LOG\_DIR=/datas/tracklogs

*## 获取昨天的日期*

yesterday=`date -d -1days '+%Y%m%d'`*####### echo "yesterday is ${yesterday}"*

*## 在此需要判断${LOG\_DIR}/${yesterday} 是否存在*

*## 循环遍历目录中的文件名称*for line in `ls ${LOG\_DIR}/${yesterday}`do

echo "loading ${line} to db\_track.yhd\_log\_load"

*## echo "prepare load the file: ${line}"*

*## 2016 11 18 19*

date=${line:0:4}${line:4:2}${line:6:2}

*# date = ${line:0:8}*

hour=${line:8:2}

*## echo "date: ${date}, hour: ${hour}"*

${HIVE\_HOME}/bin/hive -e "load data local inpath '${LOG\_DIR}/${yesterday}/${line}' into table db\_track.yhd\_log\_load partition (date = '${date}', hour = '${hour}') ;"done

#### **方式二：脚本调用sql文件**

load\_track\_logs.sh：

#!/bin/sh*## 环境变量生效*

. /etc/profile

*## HIVE HOME*

HIVE\_HOME=/opt/cdh-5.3.6/hive-0.13.1-cdh5.3.6

*## 日志目录*

LOG\_DIR=/datas/tracklogs

*## 脚本目录*

SCRIPT\_DIR=/home/beifeng

*## 获取昨天的日期*

yesterday=`date -d -1days '+%Y%m%d'`*####### echo "yesterday is ${yesterday}"*

*## 在此需要判断${LOG\_DIR}/${yesterday} 是否存在*

*## 循环遍历目录中的文件名称*for line in `ls ${LOG\_DIR}/${yesterday}`do

echo "loading ${line} to db\_track.yhd\_log\_load"

*## echo "prepare load the file: ${line}"*

*## 2016 11 18 19*

date=${line:0:4}${line:4:2}${line:6:2}

*# date = ${line:0:8}*

hour=${line:8:2}

*## echo "date: ${date}, hour: ${hour}"*

load\_file=${LOG\_DIR}/${yesterday}/${line}

${HIVE\_HOME}/bin/hive --hiveconf LOAD\_FILE\_PARAM=${load\_file} --hiveconf DATE\_PARAM=${date} --hiveconf HOUR\_PARAM=${hour} -f ${SCRIPT\_DIR}/load\_data.sql done

load\_data.sql：

load data local inpath '${hiveconf:LOAD\_FILE\_PARAM}' into table db\_track.yhd\_log\_load partition (date = '${hiveconf:DATE\_PARAM}', hour = '${hiveconf:HOUR\_PARAM}')

# hive脚本的三种执行方式

[用户1171305](https://cloud.tencent.com/developer/user/1171305)发表于[成长道路](https://cloud.tencent.com/developer/column/1724)订阅

263

1.hive控制台执行，安装了hive之后直接命令行输入hive，进入控制台。



2.hive -e "sql语句",直接输入就会有结果，sql语句根据具体情况自己书写



3.hive -f a.sql -hivevar table=people

a.sql的内容如下

set mapred.job.queue.name=default; set hive.exec.reducers.max=48; set mapred.reduce.tasks=48; set mapred.job.name=hive;

select \* from ${table};

学习笔记：

[root@cts01 ~]# echo $BASH

/bin/bash

[root@cts01 ~]# df -h

[root@cts01 ~]# vi first\_shell.sh

#!/bin/bash

#zhujie

#auto echo hello word!

#by authors liyouguang 2019

echo "hello word:"

[root@cts01 ~]# chmod o+x first\_shell.sh

[root@cts01 ~]# ./first\_shell.sh

hello word:

[root@cts01 ~]# /bin/bash first\_shell.sh 两者等价的

sh first\_shell.sh 也可以做启动的

[root@cts01 ~]# echo hello word! 执行命令 与 执行脚本之间 没有区别

执行命令，在太多的情况下，效率相当的低。

name=liyou-----------定义变量

echo "name is $name"------------引用变量

[root@cts01 ~]# /bin/bash first\_shell.sh

name is liyou

3.

echo $UID----------系统自带的，当前用户的ID。。。，

echo $PWD------显示当前目录，在哪个目录

echo $0 -------代表脚本本身的名称

echo $1 $2------

[root@cts01 ~]# sh first\_shell.sh canshuyi

name is liyou

0

/root

first\_shell.sh

Canshuyi -------$1的

4.

echo $0

echo $1

echo "The \$? is $?"

echo "The \$\* is $\*"

echo "The \$# is $#"

[root@cts01 ~]# sh first\_shell.sh canshu

name is liyou

0

/root

first\_shell.sh

canshu

The $? is 0-----------？判断上一个是佛正确，用来判断上一个命令是否执行成功

The $\* is canshu -------\*打印所有参数

The $# is 1 ----#几个参数

[root@cts01 ~]#

-用户注册登记时间 -1 用户去登记时间 -2 国际移动用户识别码 -3 手机号-4